

Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits

Vadym Kliuchnikov¹, Dmitri Maslov^{2,3}, and Michele Mosca^{4,5}

¹ *Institute for Quantum Computing, and David R. Cheriton School of Computer Science
University of Waterloo, Waterloo, Ontario, Canada*

² *National Science Foundation
Arlington, Virginia, USA*

³ *Institute for Quantum Computing, and Dept. of Physics & Astronomy
University of Waterloo, Waterloo, Ontario, Canada*

⁴ *Institute for Quantum Computing, and Dept. of Combinatorics & Optimization
University of Waterloo, Waterloo, Ontario, Canada*

⁵ *Perimeter Institute for Theoretical Physics
Waterloo, Ontario, Canada*

January 1, 2013

Abstract

We present an algorithm, along with its implementation, to approximate single-qubit unitaries using quantum circuits consisting of Clifford and T gates. In addition to meeting the known logarithmic lower bounds on the number of gates required to approximate a unitary by a quantum circuit, we give computational evidence that a very close to the best, or the best existing approximation for unitaries of the type $\text{diag}\{1, \exp(i\phi)\}$ were indeed found. In particular, the quality of our approximation is determined by the ability of the PARI software package to find the solution of a certain type of Diophantine equation, and the choice of internal parameter Delta determining the size of a computer search. We have furthermore structured our search to guarantee that our near-optimal approximations can be found for reasonable error parameters—currently, down to 10^{-17} , allowing to execute very long quantum circuits. We discuss how to improve our implementation further to handle even smaller errors (by a few orders of magnitude) that would enable the high-precision synthesis of even larger quantum algorithms.

1 Introduction

The problem of single-qubit circuit synthesis is important for efficient quantum computation. A quantum algorithm is most often described in terms of a high level circuit/procedure whose elements could be multiple-qubit transformations such as arithmetic operations (addition, multiplication, exponentiation) or special purpose transforms, such as the quantum Fourier transform (QFT). These large transforms are then decomposed into high level logical gates, such as Toffoli, Fredkin, SWAP, arbitrary two-qubit gates, etc. Those gates are further broken down into CNOT and single-qubit gates [3], and, finally, these gates are broken down into, or approximated by, circuits of gates from the available computational gate set. Motivated by the state-of-the-art methods for fault-tolerant quantum computation [1, 9, 15], we focus on the computational gate set consisting of Clifford gates and the T gate (exact synthesis is not always possible [10, 13]). The result is a logical circuit written using Clifford (for the purpose of this paper, defined as the set of Pauli-X, Y, Z gates, Phase gate, Hadamard gate, and the CNOT) and T gates.

Furthermore, in the context of fault-tolerant implementations [7, 9] Clifford gates have a relatively low implementation cost compared to the T gates [1, 8]. As a result, it is common to measure the circuit cost in terms of the number of T gates required.

The problem of approximating an arbitrary single-qubit unitary by a quantum circuit has been studied well in the relevant literature. One of the first results providing a polynomial time and polylogarithmic in the desired error solution was the Solovay-Kitaev algorithm [6]. To approximate a single-qubit unitary to within error ε , it takes $O(\log^{2.71}(1/\varepsilon))$ steps on a classical computer and the number of gates in the resulting quantum circuit it outputs is $O(\log^{3.97}(1/\varepsilon))$. The best known upper bound on the circuit size resulting from the application of the Solovay-Kitaev algorithm is $O(\log^{3+\delta}(1/\varepsilon))$, where δ can be chosen arbitrarily small [12]. A number of approaches have been developed that use additional resources in the form of ancillae, special states, classical feedback, or whose application results in a probabilistic success of having approximated a target unitary [11, 12]. These approaches improve over the resource estimates of the Solovay-Kitaev algorithm, however, fail to match the information-theoretic lower bound of $\Omega(\log(1/\varepsilon))$ for synthesizing a random unitary with precision ε . Very recently, a new single-qubit synthesis algorithm has been announced that uses at most two ancillae prepared in the state $|0\rangle$ and guarantees a logarithmic number of gates in the resulting approximation [14]; the algorithm also runs in time polynomial in $\log(1/\varepsilon)$.

The circuits produced by [14] are asymptotically optimal. While this means that the asymptotics have been settled up to a constant factor, those constant factors matter in the actual implementations. [14] reports three-qubit circuits approximating a single-qubit unitary, whereas in this paper we return to the problem of rounding off to single-qubit unitaries (as proposed in [14], Future Work) and exactly synthesizing single-qubit circuits for them based on [13]. Using only a single qubit is already an improvement by a factor of three in terms of the total number of qubits required. Apart from using no ancillae, the synthesis algorithm used in this paper and the one reported in [14] are similar in that they rely on decreasing the power of the denominator of an element of the unitary over the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ to zero as a means of synthesizing the circuit. The algorithm reported in [14] uses three-qubit two-level unitaries (which [10] shows how to synthesize in an asymptotically optimal way, through an elegant generalization of the result in [13]). Each of these two-level unitaries may require a considerable number of T gates to be implemented as a Clifford and T circuit [2, 10]. Moreover, the number of two-level gates required to reduce the denominator by $\sqrt{2}$ is at least two, since there are three non-zero entries in the matrix. We estimate that the number of T gates required to reduce the denominator by a factor of $\sqrt{2}$ by the algorithm reported in [14] may be about 20-100, whereas the experimental results reported in this paper show that the number of T gates required to reduce the denominator by a factor of $\sqrt{2}$ in the circuits reported in this paper is only about 1.6. This results in a significant practical reduction of the resources required. Furthermore, our circuits reported in this paper are generated with a computational guarantee on their quality (defined by the parameter Delta), therefore we expect that it may be difficult to optimize them further. We study how to approximate single-qubit unitaries with precisions of practical interest—as a result, our priority is the practical performance of the software implementation and bringing down the quantum resource requirements for approximations, being the T-gate counts (all other resources are minimal—e.g., we use no ancillae), that dominate the cost of the implementation of the single-qubit unitaries.

The value ε of the desired precision in approximating a single-qubit unitary affects what algorithm may be used to obtain the approximating circuit. Indeed, if the desired error is on the order of 10^{-2} , a brute force breadth first search approach may be used to compute the approximating circuit. However, breadth first search appears to run out of classical computational resources for error values below 10^{-4} . On the other hand, one may not need to approximate single-qubit unitaries to an excessively small precision. Approximation to a higher precision than needed comes at the expense of the large number of gates needed to accomplish it, and as such is not desired. We have found, via experimenting, that our software can readily handle approximation errors of 10^{-17} without compromising the quality of the output as defined by the parameter Delta. Given the desired overall logical error of about 0.1%, such an error per $\text{diag}\{1, \exp(i\phi)\}$ gate approximation allows execution of a quantum circuit containing as many as 10^{13}

single-qubit logical gates requiring approximation. This calculation is approximate, as it assumes that 10 R_Z gates are used to approximate a single qubit gate (in reality, it is at most 5), and, more importantly, that the logical errors add up, which, for most applications and approximations, is unlikely to be the case. Random and independent noise (such as that coming from approximations—and our software can be easily tuned to give a multitude of different approximations of about the same quality) scales as the square root of the sum of absolute values of all errors. Therefore, with careful approximation and design one might realistically hope to execute 10^{26} logical single-qubit gates with the overall error under 0.1%.

We further note that for the computation of this size, each of the 10^{13} single qubit gates requires an approximation by at least 150 T gates (Tables 1 and 2), each of which takes 50 units of physical resources [8] (two levels of state distillation seems reasonable for a computation of this size), therefore the total resource count is at least $7.5 * 10^{16}$. In classical terms, $7.5 * 10^{16}$ may be regarded as, approximately, a month long computation on a 200,000 MIPS processor (comparable to the best modern processors at the time of this writing). Given quantum computers are expected to be controlled by the classical computers, the operations count for classical computers may be a reasonable upper bound for the possible number of gates executable in quantum circuits.

The above discussions motivated our choice for the compromise between the time spent on the calculations and the quality of the output. In particular, we noted that we can manage errors of the practical importance, and have thus invested additional time into computing a better approximating unitary—our results are accompanied by the computational guarantee that the circuits found are as small as it were possible to obtain. Motivated by the lessons from classical compilers and Electronic Design Automation, in the scenario when an algorithm may be compiled before its execution, well-optimized implementations precomputed earlier—such as ours—result in much better practical designs.

The question of exact synthesis of quantum Clifford and T single-qubit circuits was studied in [4, 13]. Both papers report optimal implementations of the unitaries realized by the given circuits. In order to find high precision approximating circuits, they both rely on the Solovay-Kitaev algorithm. As a result, the number of gates in the approximating circuits scales as $O(\log^{3.97}(1/\varepsilon))$. Importantly, [13] reports an algorithm that optimally decomposes a unitary over $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ into a quantum Clifford and T circuit. We rely on this latter algorithm in our paper, as well as on the observation made in [13] that finding an approximating circuit is as difficult as finding the approximating unitary. Indeed, given the approximating unitary—a unitary over the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$, the optimal circuit implementing it may be found in the number of steps linear in the number of gates such a circuit contains (i.e., as fast as one may have hoped).

2 Algorithm

In this section we describe the algorithm for approximating rotations $R_Z(\phi)$:

$$\begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix}$$

by unitaries over the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$. We use notation ω for eighth root of unity, $e^{2\pi i/8}$. We also note, similarly to [14], that any single-qubit unitary can be decomposed in terms of a constant number of Hadamard gates and $R_Z(\phi)$ ([12], solution to Problem 8.1). Therefore, the ability to approximate $R_Z(\phi)$ implies the ability to approximate any single-qubit unitary.

It suffices to approximate first column of the 2×2 matrix by a unit vector with entries in the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ to approximate the entire matrix by a quantum circuit [13]. Any such vector can be written using x and y from $\mathbb{Z}[\omega]$ as $\frac{1}{2^n}(x, y)$ where x and y satisfy the following condition:

$$|x|^2 + |y|^2 = 4^n. \tag{1}$$

We define the overall quality of approximation as:

$$\sqrt{\left|\frac{x}{2^n} - e^{-i\phi/2}\right|^2 + \left|\frac{y}{2^n}\right|^2},$$

which is proportional to Frobenius distance between matrix and its approximation.

There are two main steps in our algorithm:

1. Determine x based on $e^{-i\phi/2}$ aiming to minimize the approximation error,
2. Find y such that the condition (1) is satisfied.

There are two major difficulties in this approach. Firstly, the expression determining the quality of approximation is relatively complicated. Secondly, the requirement to satisfy condition (1) results in a system of Diophantine equations that does not always have a solution.

Now we concentrate on the solution to the first problem. We treat the power of 2 in the denominator as the input parameter to our algorithm. It is not difficult to observe that we can rewrite the square of the expression for the quality of approximation as follows:

$$\left|\frac{x}{2^n} - e^{-i\phi/2}\right|^2 + 1 - \left|\frac{x}{2^n}\right|^2.$$

As such, the problem is reduced to finding a suitable x .

We find x using brute force search procedure. As (1) does not always have a solution we build a list of approximations of x sorted by the achieved accuracy of approximation. Then we try to solve (1) in the order of decreasing quality of the approximation. Once we succeed we output the unitary and produce its decomposition into a circuit relying on the exact synthesizer reported in [13].

In more details, to build a list of approximations of x , we first build lists RE and IM of approximations of $\Re(2^n e^{-i\phi/2})$ and $\Im(2^n e^{-i\phi/2})$, real and imaginary parts, by numbers of the form $a + \sqrt{2}b$ with quality of approximation at least $2^{-\text{Delta}}$, limiting $|a| < 2^n$ and $|b| < 2^n$. Next we compute quantities:

$$\begin{aligned} D_{\Re} &= 4^n \cos^2(\phi/2) - |a + \sqrt{2}b|^2 + |2^n \cos(\phi/2) - (a + \sqrt{2}b)|^2 \\ D_{\Im} &= 4^n \sin^2(\phi/2) - |c + \sqrt{2}d|^2 + |(c + \sqrt{2}d) - 2^n \sin(\phi/2)|^2, \end{aligned}$$

where $a + \sqrt{2}b$ approximates $\Re(2^n e^{-i\phi/2})$ and $c + \sqrt{2}d$ approximates $\Im(2^n e^{-i\phi/2})$. Note that D_{\Re} and D_{\Im} sums together to the square of the overall quality of approximation that we are aiming to minimize multiplied by 4^n . Having D_{\Re} and D_{\Im} precomputed allows us to efficiently build a list of elements with overall quality of approximation below the selected threshold β . We sort lists of D_{\Re} and D_{\Im} . For each element α of D_{\Re} we look up all elements of D_{\Im} in the range $[\alpha - \beta, \alpha + \beta]$ using binary search. We found empirically that threshold value of 1 results in the list with size approximately equal to the number of elements in lists RE and IM . If we do not find a solution of the Diophantine equation for elements in the list we increase the threshold and consider elements with lower quality of approximation.

To find y on the second step of the algorithm, we are already given x , that was obtained on the first step. Equation (1) can be written as:

$$|y|^2 = C_1 + \sqrt{2}C_2. \tag{2}$$

We find y via a reduction of the above equation to a well studied problem in algebraic number theory. We first introduce the required notations. The norm of an element $a + \sqrt{2}b$ of the ring $\mathbb{Z}[\sqrt{2}]$ is defined as follows:

$$N_{\sqrt{2}}(a + \sqrt{2}b) = a^2 - 2b^2$$

In other words, it is computed by the multiplication of $a + \sqrt{2}b$ by its adjoint, $a - \sqrt{2}b$. We also recall the definition of the norm of an element z in $\mathbb{Z}[\omega]$. Note that $|z|^2$ belongs to $\mathbb{Z}[\sqrt{2}]$. The norm itself is defined as:

$$N(z) = N_{\sqrt{2}}(|z|^2).$$

It is important to remember that the norm is a multiplicative function of the ring elements.

The necessary condition that any solution of (1) must satisfy is:

$$N(y) = N_{\sqrt{2}}(|y|^2) = N_{\sqrt{2}}(C_1 + \sqrt{2}C_2).$$

The problem of finding a solution to the above equation is a known problem in algebraic number theory [18]. Even more, there exists an efficient algorithm that solves it for general extension rings. It is included in the open source number-theoretic library PARI [16] that we use as a part of our software implementation for unitary approximations.

We next describe how to find a solution to our problem given the solution to the norm equation. There are two main issues here. The first is that the norm of the number is invariant with respect to the multiplication by units in the ring, being those elements with norm equal to 1. The second is that it is invariant with respect to taking the adjoint:

$$N_{\sqrt{2}}(C_1 + \sqrt{2}C_2) = N_{\sqrt{2}}(C_1 - \sqrt{2}C_2)$$

Therefore, once we find y that is a solution to the norm equation, we check if $|y|^2$ divides $C_1 + \sqrt{2}C_2$. If it does not, we perform the following transformation:

$$y = a + b\omega + c\omega^2 + d\omega^3 \mapsto a - b\omega + c\omega^2 - d\omega^3,$$

that effectively changes $|y|^2$ to its adjoint in the ring $\mathbb{Z}[\sqrt{2}]$. In some rare cases $|y|^2$ does not divide both $C_1 + \sqrt{2}C_2$ and $C_1 - \sqrt{2}C_2$. We have not investigated the reason for this, and simply skip such an approximation of x and try to solve the equation for a different approximation of x .

Once we successfully found y such that $|y|^2$ divides $C_1 + \sqrt{2}C_2$, this means that we expressed

$$u|y|^2 = (C_1 + \sqrt{2}C_2),$$

for u being a unit in the ring $\mathbb{Z}[\sqrt{2}]$. It is well known fact that units of $\mathbb{Z}[\sqrt{2}]$ are of the form $(-1)^k(\sqrt{2} \pm 1)^m$. Therefore, we just need to find k and m corresponding to u . Given k and m we find \sqrt{u} , if it is defined, and obtain the solution to the equation:

$$|y\sqrt{u}|^2 = C_1 + \sqrt{2}C_2.$$

We use a simple algorithm to find k and m in the expression for unit u . We divide u by the powers of $\sqrt{2} \pm 1$ and check if the result, written in the form $A + \sqrt{2}B$, has the absolute value of both coefficients A and B that are strictly less than the absolute value of the corresponding coefficients of u . This means that u included the corresponding power of basic units. We repeat this process until we get a 1 or -1 .

3 Experimental results

We report experiments with the $R_Z(1/10)$, that we used to demonstrate the scaling of our approach. Future revisions will contain an expanded set of results. For our experiments we used a high performance server with eight Quad-Core AMD Opteron 8356 (2.30 GHz) processors and 128 GB of RAM memory. Our current algorithm implementation completely utilizes the processing power of the server and runs 32 threads in parallel.

Table 1 summarizes our results. The first column reports the power of 2 in the denominator of the approximating unitary. Second column reports the value of the parameter Delta that we use to adjust the size of the search space. The T-gate count shows the number of T gates required in the circuit implementing the approximating unitary (synthesized using [13]). We use global phase invariant metric (Formula 1 in

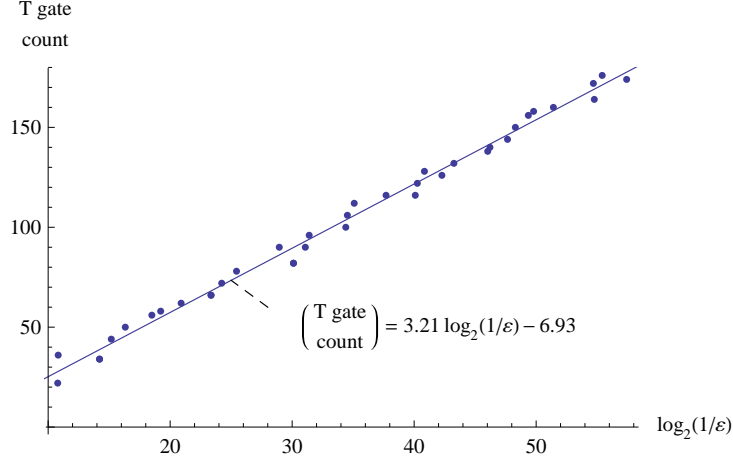


Figure 1: The dependency of T-count on $\log_2(1/\epsilon)$. With confidence level 0.999, the slope coefficient belongs to the interval $[3.04, 3.38]$ and the additive constant belongs to the interval $[-13.34, -0.51]$.

Power of denominator	Delta (2^{-m})	T gate count	Precision	Approximation time (s)	PARI time (s)
8	4	22	5.68E-04	0.0186	1.0823
9	4	36	5.52E-04	0.0168	22.132
10	5	34	5.26E-05	0.0254	0.6993
11	5	34	5.26E-05	0.0403	6.7426
12	6	44	2.70E-05	0.0288	10.367
13	6	50	1.22E-05	0.0370	18.206
14	7	56	2.70E-06	0.0523	3.8686
15	7	58	1.63E-06	0.0603	11.638
16	8	62	5.11E-07	0.0759	5.0739
17	8	66	9.34E-08	0.0896	0.8770
18	9	66	9.34E-08	0.1086	7.1461
19	9	72	5.10E-08	0.2008	19.890
20	10	78	2.20E-08	0.1909	17.955
21	10	82	8.63E-10	0.4383	0.3041
22	11	82	8.63E-10	0.4311	1.6429
23	11	90	1.94E-09	0.8516	35.749
24	12	90	4.40E-10	0.9669	8.2583
25	12	96	3.54E-10	1.7322	45.083
26	13	100	4.43E-11	1.8963	3.9753
27	13	106	4.04E-11	4.1780	20.981
28	14	112	2.74E-11	4.2127	40.875
29	14	116	8.53E-13	7.5655	0.8634
30	15	116	4.49E-12	8.4449	36.773
31	15	122	7.59E-13	16.947	7.6613
32	16	128	5.07E-13	18.060	16.718
33	16	126	1.88E-13	35.986	19.828
34	17	132	9.50E-14	40.933	20.558
35	17	138	1.39E-14	79.948	4.2597
36	18	140	1.23E-14	98.815	14.295
37	18	144	4.52E-15	188.24	14.698
38	19	150	2.89E-15	244.19	22.326
39	19	156	1.38E-15	477.75	41.527
40	20	158	1.03E-15	658.85	97.711
41	20	160	3.33E-16	1295.0	83.670
42	21	164	3.26E-17	1890.6	3.7347
43	21	172	3.44E-17	3858.4	30.974
44	22	176	2.08E-17	6307.2	45.898
45	22	174	5.19E-18	12614.5	24.259

Table 1: The results of approximating $R_Z(1/10)$. For n , the power of 2 in the denominator, Delta was chosen to be $\lfloor n/2 \rfloor$.

Power of denominator	Delta (2^{-m})	T gate count	Precision	Approximation time (s)	PARI time (s)
19	4	74	4.99E-08	2.3301	27.379
19	9	72	5.10E-08	0.2008	19.890
20	5	78	1.94E-08	2.4479	26.532
20	10	78	2.20E-08	0.1909	17.955
23	5	88	7.36E-10	22.615	5.7688
23	11	90	1.94E-09	0.8516	35.749
24	6	94	2.46E-10	28.278	3.8146
24	12	90	4.40E-10	0.9669	8.2583
25	6	100	1.60E-10	53.710	15.435
25	12	96	3.54E-10	1.7322	45.083
27	6	106	1.61E-11	211.62	4.2888
27	13	106	4.04E-11	4.1780	20.981
28	7	110	1.49E-11	242.06	51.418
28	14	112	2.74E-11	4.2127	40.875
30	7	116	8.53E-13	975.14	2.4162
30	15	116	4.49E-12	8.4449	36.773
31	7	120	7.49E-13	2000.3	27.550
31	15	122	7.59E-13	16.947	7.6613
32	8	124	1.06E-13	2359.0	1.4174
32	16	128	5.07E-13	18.060	16.718
33	11	124	1.06E-13	656.84	19.465
33	16	126	1.88E-13	35.986	19.828
34	11	130	2.19E-14	1481.0	2.4933
34	17	132	9.50E-14	40.933	20.558
35	11	136	6.72E-15	3111.2	1.5797
35	17	138	1.39E-14	79.948	4.2597

Table 2: Approximations of $R_Z(1/10)$. Included are the cases when decreasing Delta resulted in better approximations.

[7]) to measure the achieved precision. The following two columns report the runtime, in seconds, of the most extensive parts of our computation—the time it took to build a list of approximations of $e^{-i\phi/2}$, and the time it took for PARI to solve the norm equation.

Table 2 illustrates how increasing the size of the search space (decreasing the value of the parameter Delta) influences the performance of our algorithm and the quality of the results. The table shows only those cases when increasing the size of search space resulted in a better approximation. In our baseline experiment we used Delta equal to $\lfloor n/2 \rfloor$, where n is the power of 2 in the denominator. We also performed experiments with Delta equal to $\lfloor n/3 \rfloor$ and $\lfloor n/4 \rfloor$. With this choice of Delta we stopped the experiments at $n = 35$ and $n = 32$, correspondingly, as the available memory became a limitation. This is a limitation of the current implementation and can be solved by redesign of our software.

Table 3 shows the comparison of our results to those recently published in [17]. Our circuits feature the T-count smaller by about 22-30%, which is important in practice.

4 Future work

Our immediate research plans include software optimization, with the goal of improving the balance between computational speed and quality of approximation, as well as tighter implementation of our

Results reported in [17]		Our results achieving precision at least ε			Our results using at most N gates	
T gate count (N)	Precision (ε)	T gate count	Precision	Reduction of T gate count (%)	T gate count	Precision
78	3.17E-06	56	2.70E-06	28	74	4.99E-08
144	6.85E-11	100	4.43E-11	30	136	6.72E-15
210	6.28E-16	160	3.33E-16	23	174	5.19E-18

Table 3: Comparison to the results reported in [17]. The precision is reported using Fowler’s metric [7], but the comparisons are essentially the same using other metrics.

algorithms. Our algorithm is highly parallelizable, and it can benefit greatly from the execution on the parallel hardware. We expect that the aggregate result of software optimizations and parallelization will help decrease the error parameter for which we are able to search approximations with the computational quality guarantee by a few to, possibly, several orders of magnitude. We further plan to report extensive benchmark results, most importantly, using $R_Z(\frac{1}{2^k})$ single-qubit gates that are common in practical applications. While we employ an expensive algorithm for finding coordinate approximations, and faster approximating algorithms are possible—e.g., based on continued fractions approximation, those faster algorithms may compromise the guarantee on the quality of the resulting approximating circuits. On the other hand, it may be expected that those algorithms are able to handle much smaller errors.

5 Conclusion

In this paper, we reported an algorithm for construction of approximating circuits for single-qubit unitaries. We targeted practical error sizes, and structured our search as to provide a form of exhaustive computational guarantee on the quality of the final result. The quality of circuits/approximations is defined by the parameter Delta. The error sizes we can handle, down to 10^{-17} , appear to be practical. Furthermore, our future optimizations are expected to make it possible to scale further.

6 Acknowledgements

Authors supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center Contract number D11PC20166. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC or the U.S. Government.

This material is based upon work partially supported by the National Science Foundation (NSF), during D. Maslov’s assignment at the Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Michele Mosca is also supported by Canada’s NSERC, MPrime, CIFAR, and CFI. IQC and Perimeter Institute are supported in part by the Government of Canada and the Province of Ontario.

We wish to thank Martin Roetteler for many helpful discussions.

References

- [1] P. Aliferis, D. Gottesman, and J. Preskill. *Quantum accuracy threshold for concatenated distance-3 codes*. Quantum Information and Computation, 6:97–165, 2006, [quant-ph/0504218](#).
- [2] M. Amy, D. Maslov, M. Mosca, and M. Roetteler. *A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits*. 2012, [arXiv:1206.0758](#).
- [3] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. *Elementary gates for quantum computation*. Physical Review A 52, 3457–3467, 1995, [quant-ph/9503016](#).
- [4] A. Bocharov and K. M. Svore. *A depth-optimal canonical form for single-qubit quantum circuits*. 2012, [arXiv:1206.3223](#).
- [5] H. K. Cummins, G. Llewellyn, and J. A. Jones. *Tackling Systematic Errors in Quantum Logic Gates with Composite Rotations*. Physical Review A 67:042308, 2003, [quant-ph/0208092](#).
- [6] C. Dawson and M. Nielsen. *The Solovay-Kitaev algorithm*. Quantum Information and Computation 6:81–95, 2006, [quant-ph/0505030](#).
- [7] A. G. Fowler. *Constructing Arbitrary Steane Code Single Logical Qubit Fault-tolerant Gates*. Quantum Information and Computation 11:867–873, 2011, [quant-ph/0411206](#).
- [8] A. G. Fowler and S. J. Devitt. *A bridge to lower overhead quantum computation*, September 2012, [arXiv:1209.0510](#).
- [9] A. G. Fowler, A. Stephens, and P. Groszkowski. *High threshold universal quantum computation on the surface code*. Physical Review A 80, 052312, November 2009, [arXiv:0803.0272v4](#).
- [10] B. Giles and P. Selinger. *Exact synthesis of multi-qubit Clifford+T circuits*, December 2012, [arXiv:1212.0506](#).
- [11] N. C. Jones, J. D. Whitfield, P. L. McMahon, M.-h. Yung, R. Van Meter, A. Aspuru-Guzik, and Y. Yamamoto. *Simulating chemistry efficiently on fault-tolerant quantum computers*. April 2012, [arXiv:1204.0567v1](#).
- [12] A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, Providence, RI, 2002.
- [13] V. Kliuchnikov, D. Maslov, and M. Mosca. *Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates*. 2012, [arXiv:1206.5236](#).
- [14] V. Kliuchnikov, D. Maslov, and M. Mosca. *Asymptotically optimal approximation of single qubit unitaries by Clifford and T circuits using a constant number of ancillary qubits*. 2012, [arXiv:1212.0822](#).
- [15] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [16] PARI, a computer algebra system, Online: <http://pari.math.u-bordeaux.fr>, downloaded December 2012.
- [17] P. Selinger. *Efficient Clifford+T approximation of single-qubit operators*. December 2012, [arXiv:1212.6253](#).
- [18] D. Simon. *Solving norm equations in relative number fields using S-units*. Mathematics of Computation 239:1287–1305, 2002.